

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-282674

(43)Date of publication of application : 15.10.1999

(51)Int.Cl.

G06F 9/30
G06F 9/30
G06F 9/32
G06F 9/38
G06F 9/38

(21)Application number : 10-083368

(71)Applicant : MATSUSHITA ELECTRIC IND CO LTD

(22)Date of filing : 30.03.1998

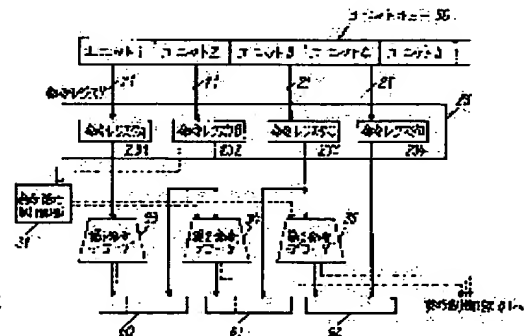
(72)Inventor : HEIJI TAKEHITO
TANAKA TETSUYA
HIGAKI NOBUO
TAKAYAMA SHUICHI
KOTANI KENSUKE

(54) PROCESSOR

(57)Abstract:

PROBLEM TO BE SOLVED: To reduce the code size by using one or plural word elements to form a unit instruction serving as an execution unit and using two types of these word elements.

SOLUTION: An instruction issue control part 31 controls an instruction decoder by making reference to the format information as well as the parallel execution boundary information on the units which are stored in the instruction registers A231 and B232. Each of instruction decoders 33 to 35 inputs a unit of 21 bits to decode it, outputs the control signal concerning the operation of an instruction consisting of the said unit to an execution control part 41 and also outputs a constant operand included in the unit. In this case, the specific one of instruction registers 23 to which a certain unit is transferred is uniquely decided based on the position (order) of a unit queue 50. In other words, the units 1 and 2 are transferred to the registers A231 and B232 respectively. Thus, a selector of units is not required.



LEGAL STATUS

[Date of request for examination] 02.03.2001

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]	3451921
[Date of registration]	18.07.2003
[Number of appeal against examiner's decision of rejection]	
[Date of requesting appeal against examiner's decision of rejection]	
[Date of extinction of right]	

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-282674

(43) 公開日 平成11年(1999)10月15日

(51) IntCl.⁹

識別記号

F I

G 0 6 F 9/30

3 5 0

G 0 6 F 9/30

3 5 0 F

3 1 0

3 1 0 C

9/32

3 5 0

9/32

3 5 0 A

9/38

3 1 0

9/38

3 1 0 H

3 7 0

3 7 0 X

審査請求 未請求 請求項の数12 O L (全 19 頁)

(21) 出願番号

特願平10-83368

(22) 出願日

平成10年(1998) 3 月30日

(71) 出願人 000005821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(72) 発明者 瓶子 岳人

大阪府門真市大字門真1006番地 松下電器

産業株式会社内

(72) 発明者 田中 哲也

大阪府門真市大字門真1006番地 松下電器

産業株式会社内

(72) 発明者 檜垣 信生

大阪府門真市大字門真1006番地 松下電器

産業株式会社内

(74) 代理人 弁理士 滝本 智之 (外 1 名)

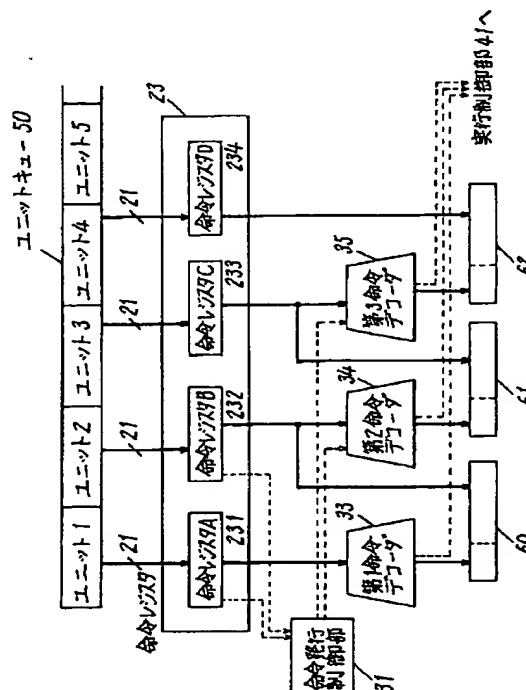
最終頁に続く

(54) 【発明の名称】 プロセッサ

(57) 【要約】

【課題】 VLIW方式のような静的な並列スケジューリングを前提としたプロセッサにおいて、コードサイズの削減を図る。また、可変長命令方式におけるハードウェア複雑化の問題点を克服し、性能向上とコード効率向上の両立を図る。

【解決手段】 命令構成要素内にフォーマット情報を持ち、それを参照することによりいくつかの命令構成要素を連結して一つの命令を構成するのかを決定する。ハードウェア簡単化のため、命令構成要素内のフォーマット情報を参照することにより、後続する命令構成要素の命令としての解釈を無効化する機構を備える。



【特許請求の範囲】

【請求項 1】 一語が複数でかつ所定数の語素からなる複合命令を実行する V L I W 方式のプロセッサであって、

実行の単位となる単位命令が一または複数の前記語素から構成され、かつ前記単位命令を構成する前記語素の数が少なくとも二通りあることを特徴とするプロセッサ。

【請求項 2】 前記単位命令を構成する前記語素の数に関する語数情報が、前記単位命令内に明示的に含まれることを特徴とする請求項 1 記載のプロセッサ。

【請求項 3】 一語が複数の語素からなる複合命令を実行し、実行の単位となる単位命令が一または複数の前記語素から構成され、かつ前記単位命令を構成する前記語素の数が少なくとも二通りあるプロセッサであって、記憶装置から前記複合命令を読み出す複合命令読み出し手段と、

前記複合命令読み出し手段により読み出された前記複合命令中の前記語素から、いずれかの前記語素内に含まれる前記単位命令を構成する前記語素の数に関する語数情報に基づいて、前記単位命令を組み立てて解読する解読手段とを備えることを特徴とするプロセッサ。

【請求項 4】 前記解読手段は、前記語素の一定数ごとに区切られた前記複合命令をそれぞれ解読する複数の単位解読手段と、前記語数情報に基づいて前記単位解読手段のいくつかの解読を無効化する解読制御手段とから構成されることを特徴とする請求項 3 記載のプロセッサ。

【請求項 5】 前記解読制御手段は、前記語数情報に基づいて、命令中のオペレーションに関する記述を解読していない前記単位解読手段に対して前記無効化を行うことを特徴とする請求項 4 記載のプロセッサ。

【請求項 6】 前記一定数は、前記単位命令を構成する前記語素の最小の数であることを特徴とする請求項 5 記載のプロセッサ。

【請求項 7】 前記プロセッサは、一語が複数でかつ所定数の語素からなる複合命令を実行する V L I W 方式のプロセッサであって、さらに、前記複合命令が一語に含み得る前記単位命令の数に等しい数の、前記単位命令を実行する実行手段を備えることを特徴とする請求項 3 から 6 のいずれか 1 項に記載のプロセッサ。

【請求項 8】 前記単位命令は、第 1 の数の前記語素から構成される単位命令と、前記第 1 の数より大きい第 2 の数の前記語素から構成される単位命令とがあり、前記第 2 の数の前記語素から構成される単位命令の内にあって、前記第 1 の数の前記語素から構成される単位命令に相当する部分からはみ出す部分の語素には、命令中の数値に関する記述のみが配置されることを特徴とする請求項 1 から 7 のいずれか 1 項に記載のプロセッサ。

【請求項 9】 前記複合命令中の前記単位命令と該単位

命令に後続する単位命令とが並列に実行できるか否かを示す並列実行情報が、前記複合命令内に明示的に含まれることを特徴とする請求項 1 または 2 記載のプロセッサ。

【請求項 10】 前記並列実行情報が、並列に実行できる前記単位命令の群と後続する前記群との境界に関する情報であることを特徴とする請求項 9 記載のプロセッサ。

【請求項 11】 前記プロセッサは、記憶装置から前記複合命令を読み出す複合命令読み出し手段と、

前記複合命令読み出し手段により読み出された前記複合命令中の前記語素から、前記語数情報と前記並列実行情報とに基づいて前記命令単位を組み立てて解読する解読手段とを備えることを特徴とする請求項 9 または 10 記載のプロセッサ。

【請求項 12】 前記解読手段は、前記語素の一定数ごとに区切られた前記複合命令をそれぞれ解読する複数の単位解読手段と、前記語数情報と前記並列実行情報とに基づいて前記単位解読手段のいくつかの解読を無効化する解読制御手段とから構成されることを特徴とする請求項 11 記載のプロセッサ。

【発明の詳細な説明】**【0001】**

【発明の属する技術分野】本発明は、プロセッサに関し、特に並列処理による実行サイクル数の削減とコード効率の向上を図る技術に関する。

【0002】

【従来の技術】近年のマイクロプロセッサ応用製品の高機能化及び高速化に伴い、高い処理性能を持つマイクロプロセッサ（以下、単に「プロセッサ」という。）が望まれている。そして、これを実現する技術の 1 つとして 1 サイクルに複数の命令を同時に実行するものがあり、V L I W (Very Long Instruction Word) 方式のプロセッサもその技術の 1 つである。

【0003】この V L I W 方式のプロセッサは、実行コード生成時にコンパイラ等により静的に命令間の依存関係を解析し、命令コードの移動を行って実行効率の良い命令ストリームを生成するものである。この方式は、動的に命令間の依存関係を解析するスーパースカラ方式のプロセッサと比べてハードウェアを簡略化でき、このため動作周波数を上げやすいという長所を有する。

【0004】しかし、V L I W 方式では、一般に命令を固定長として取り扱うため、次のような問題がある。

【0005】すなわち、長い定数を扱う命令の指定には多くのビット数を必要とするが、レジスタ間演算命令の指定にはそれほど多くのビット数を必要としない等、命令毎に指定に必要なビット数にばらつきがある。ところが、V L I W 方式では命令を固定長として取り扱うた

め、短いビット数しか必要としない命令を指定するのに必要以上のビット数を用いざるを得ず、コードサイズが大きくなってしまふ。

【0006】そして、この問題を解決する1つの手段として、命令長を可変とすることが考えられる。

【0007】図13は、1つの命令が1個または2個の命令構成要素（ここでは「ユニット」と呼ぶ）にて構成され、3つの命令を同時実行可能なプロセッサの命令レジスタ周辺の構成を示すブロック図である。同図において、破線は制御信号を表している。図13においてユニットキュー50は、ユニットの並びであり、命令メモリ等から供給された順にユニットを命令レジスタに転送していく。

【0008】この構成では、命令レジスタA52aと命令レジスタB52b、命令レジスタC52cと命令レジスタD52d、命令レジスタE52eと命令レジスタF52fがそれぞれ対になっており、命令は常に命令レジスタA52a、命令レジスタC52c又は命令レジスタE52eの3つのレジスタのいずれかを先頭として命令レジスタに格納され、2つのユニットを連結して1つの命令を構成する場合にのみ、対となっているもう一方の命令レジスタにユニットが転送される。したがって、命令レジスタA52aに転送されたユニットがそのユニット単体で命令を構成する場合には、命令レジスタB52bにはユニットが転送されないことになる。

【0009】図13を見るとわかるように、この構成ではユニットキュー50の各ユニットがいずれの命令レジスタに転送されるのかが一意に決まっていない。また、各命令レジスタへ転送されるユニットがユニットキュー50のいずれのユニットなのかが一意に決まっていない。そこで、セクタ51a～51dを制御して転送するユニットを選択することになる。さらに、これらのセクタの制御は全体を一度に決定することができず、まずセクタ51a、セクタ51bの制御が決定され、命令レジスタCへ転送されるユニットが決定してから、このユニット内の命令長に関する情報を参照して、図中破線で示すようにセクタ51c、セクタ51dの制御を決定する。

【0010】

【発明が解決しようとする課題】しかしながら、上記従来のプロセッサでは、ユニットキューから命令レジスタへの転送の際の遅延が大幅に大きくなるという問題点があった。これは、先行する命令レジスタに転送されたユニット内の命令長に関する情報を参照しなければ、当該命令レジスタに関するセクタの制御を決定できないからである。また、並列度が増すに従って、転送すべき命令レジスタの数が増加するので、この遅延はさらに大きくなっていく。

【0011】一方、ユニットキュー内のユニットと命令レジスタとの対応を一对一とし、図13に示したプロセ

ッサで問題となっていたユニットキューから命令レジスタへの転送におけるセクタによる遅延の問題を解消する技術として、図14に示すものがある。このプロセッサでは、命令を構成する可能性のある全てのユニットの組み合わせについてデコードしておき、先行する命令デコーダから出力される命令長の情報によって、デコード結果を選択して使用する。具体的には、図中破線で示すように、第1命令デコーダ53dが出力する情報によりセクタ51eの制御を決定し、上記情報と第2命令デコーダ53eまたは第3命令デコーダ53fが出力する情報によりセクタ51fの制御を決定する。

【0012】ところが、3つの命令を同時実行するために、2ユニット長の命令を解釈するデコーダを5つも必要とし、ハードウェアが非常に大きくなるという問題点がある。

【0013】そこで、本発明はかかる問題点を鑑みてなされたものであり、命令レベルの並列実行に際して、ハードウェア複雑化の問題を克服しつつ、性能向上とコード効率向上を両立するプロセッサを提供することを目的とする。

【0014】

【課題を解決するための手段】上記目的を達成するために請求項1記載のプロセッサは、一語が複数でかつ所定数の語素からなる複合命令を実行するVLIW方式のプロセッサであって、実行の単位となる単位命令が一または複数の前記語素から構成され、かつ前記単位命令を構成する前記語素の数が少なくとも二通りあることを特徴とする。

【0015】これによって、各命令の指定に必要なビット数に応じた命令長の命令フォーマットにて命令を指定することができ、コードサイズを削減することができる。

【0016】ここで請求項2記載のプロセッサは、請求項1記載のプロセッサにおいて、前記単位命令を構成する前記語素の数に関する語数情報が、前記単位命令内に明示的に含まれることを特徴とする。

【0017】これによって、命令の発行制御部において、命令長を特定することが容易になり、命令発行制御を行うハードウェアを単純化することができる。

【0018】また上記目的を達成するために請求項3記載のプロセッサは、一語が複数の語素からなる複合命令を実行し、実行の単位となる単位命令が一または複数の前記語素から構成され、かつ前記単位命令を構成する前記語素の数が少なくとも二通りあるプロセッサであって、記憶装置から前記複合命令を読み出す複合命令読み出し手段と、前記複合命令読み出し手段により読み出された前記複合命令中の前記語素から、いずれかの前記語素内に含まれる前記単位命令を構成する前記語素の数に関する語数情報に基づいて、前記単位命令を組み立てて解釈する解釈手段とを備えることを特徴とする。

【0019】これによって、各命令の指定に必要なビット数に応じた命令長の命令フォーマットにて命令を指定することができ、コードサイズを削減することができる。

【0020】また請求項4記載のプロセッサは、請求項3記載のプロセッサにおいて、前記解読手段は、前記語素の一定数ごとに区切られた前記複合命令をそれぞれ解読する複数の単位解読手段と、前記語数情報に基づいて前記単位解読手段のいくつかの解読を無効化する解読制御手段とから構成されることを特徴とする。

【0021】これによって、命令の発行部のハードウェアを単純化することができ、命令長を可変とすることによりハードウェアが複雑になり、遅延が増大して性能が低下するという問題点を克服している。

【0022】また請求項5記載のプロセッサは、請求項4記載のプロセッサにおいて、前記解読制御手段は、前記語数情報に基づいて、命令中のオペレーションに関する記述を解読していない前記単位解読手段に対して前記無効化を行うことを特徴とする。

【0023】これによって、命令の発行部のハードウェアを単純化することができ、命令長を可変とすることによりハードウェアが複雑になり、遅延が増大して性能が低下する、という問題点を克服している。

【0024】さらに請求項6記載のプロセッサは、請求項5記載のプロセッサにおいて、前記一定数は、前記単位命令を構成する前記語素の最小の数であることを特徴とする。

【0025】また請求項7記載のプロセッサは、請求項3から請求項6のいずれか1項に記載のプロセッサにおいて、一語が複数でかつ所定数の語素からなる複合命令を実行するVLIW方式のプロセッサであって、さらに、前記複合命令が一語に含み得る前記単位命令の数に等しい数の、前記単位命令を実行する実行手段を備えることを特徴とする。

【0026】また請求項8記載のプロセッサは、請求項1から請求項7のいずれか1項に記載のプロセッサにおいて、前記単位命令は、第1の数の前記語素から構成される単位命令と、前記第1の数より大きい第2の数の前記語素から構成される単位命令とがあり、前記第2の数の前記語素から構成される単位命令の内にあって、前記第1の数の前記語素から構成される単位命令に相当する部分からはみ出す部分の語素には、命令中の数値に関する記述のみが配置されることを特徴とする。

【0027】これによって、命令解読器の入力ビット幅を短くすることができ、ハードウェアを単純化することができる。

【0028】さらに請求項9記載のプロセッサは、請求項1または請求項2記載のプロセッサにおいて、前記複合命令中の前記単位命令と該単位命令に後続する単位命令とが並列に実行できるか否かを示す並列実行情報が、

前記複合命令内に明示的に含まれることを特徴とする。

【0029】これによって、各命令の指定に必要なビット数に応じた命令長の命令フォーマットにて命令を指定することができると共に、一般のVLIWなどで並列実行可能な命令が存在しない場合に挿入される無動作命令の挿入が不要となり、コードサイズを削減することができる。

【0030】また請求項10記載のプロセッサは、請求項9記載のプロセッサにおいて、前記並列実行情報が、並列に実行できる前記単位命令の群と後続する前記群との境界に関する情報であることを特徴とする。

【0031】さらに請求項11記載のプロセッサは、請求項9または請求項10記載のプロセッサにおいて、記憶装置から前記複合命令を読み出す複合命令読み出し手段と、前記複合命令読み出し手段により読み出された前記複合命令中の前記語素から、前記語数情報と前記並列実行情報とに基づいて前記命令単位を組み立てて解読する解読手段とを備えることを特徴とする。

【0032】これによって、各命令の指定に必要なビット数に応じた命令長の命令フォーマットにて命令を指定することができると共に、一般のVLIWなどで並列実行可能な命令が存在しない場合に挿入される無動作命令の挿入が不要となり、コードサイズを削減することができる。

【0033】また請求項12記載のプロセッサは、請求項11記載のプロセッサにおいて、前記解読手段は、前記語素の一定数ごとに区切られた前記複合命令をそれぞれ解読する複数の単位解読手段と、前記語数情報と前記並列実行情報とに基づいて前記単位解読手段のいくつかの解読を無効化する解読制御手段とから構成されることを特徴とする。

【0034】これによって、各命令の指定に必要なビット数に応じた命令長の命令フォーマットにて命令を指定することができると共に、一般のVLIWなどで並列実行可能な命令が存在しない場合に挿入される無動作命令の挿入が不要となり、コードサイズを削減することができる。さらに、命令の発行部のハードウェアを単純化することができ、命令長を可変とすることによりハードウェアが複雑になり、遅延が増大して性能が低下するという問題点を克服している。

【0035】

【発明の実施の形態】以下、本発明に係るプロセッサの実施の形態について、図面を用いて詳細に説明する。

（命令フォーマットとアーキテクチャの概要）まず、本プロセッサが解読実行する命令（特許請求の範囲に記載する「単位命令」に相当する。）の構造について説明する。

【0036】図1（a）～図1（e）は本プロセッサの命令フォーマットを示す図である。本プロセッサの各命令は、21ビットの命令構成要素（ここでは「ユニッ

ト」と呼ぶ。特許請求の範囲に記載する「語素」に相当する。)にて構成されており、ユニット1つで構成される21ビット命令とユニット2つで構成される42ビット命令の2種類の命令フォーマットが存在する。各命令がいずれの長さの命令であるかは、1ビットのフォーマット情報11によって決定される。具体的には、フォーマット情報11が“0”の時はそのユニット単体で21ビット命令となり、フォーマット情報11が“1”の時はそのユニットとそれに後続するユニットとを連結して42ビット命令となる。

【0037】また、各命令には1ビットの並列実行境界情報10を持たせてある。この情報は、この命令とそれに後続する命令との間に並列実行の境界が存在するか否かを示すものである。具体的には、並列実行境界情報10が“1”の時はその命令と後続命令の間に並列実行の境界が存在し、並列実行境界情報10が“0”の時には並列実行の境界が存在しないことになる。この情報の利用方法については後で述べる。

【0038】各命令の命令長からフォーマット情報11と並列実行境界情報10を除いた部分にてオペレーションを指定する。21ビット命令では19ビット、42ビット命令では40ビットの長さを使用することができることになる。具体的には、“Op1”、“Op2”、“Op3”のフィールドでは、オペレーションの種類を表すオペコードを、“Rs”のフィールドでは、ソースオペランドとなるレジスタのレジスタ番号を、“Rd”のフィールドでは、デスティネーションオペランドとなるレジスタのレジスタ番号を指定する。また、“imm5”及び“imm32”のフィールドでは、それぞれ5ビットと32ビットの演算用定数オペランドを指定する。そして、“disp13”及び“disp31”のフィールドでは、それぞれ13ビットと31ビットの変位(ディスプレースメント)を指定する。

【0039】32ビットの定数などの長い定数を扱う転送命令や演算命令、大きなディスプレースメントを指定する分岐命令は42ビット命令で定義され、それらを除くほとんどの命令は21ビット命令で定義されている。なお、図1を見てわかるように、42ビット命令の構成要素である2つのユニットのうち、後ろの方のユニットすなわち2番目のユニットには、長い定数やディスプレースメントの一部のみが配置され、オペコードは配置されない。

【0040】次に、本プロセッサのアーキテクチャの概要について説明する。本プロセッサは、静的な並列スケジューリングを前提としたプロセッサであって、命令の供給と発行の概念は図2のようになる。

【0041】命令の供給は、同図(a)に示すように毎サイクル64ビット固定長の命令供給単位(ここでは「パケット」と呼ぶ。特許請求の範囲に記載する「複合命令」に相当する。)でユニットを3個ずつ供給する。

ユニット3個分の長さは63ビットであるが、残りの1ビットについては使用しない。そして、命令の実行は、同図(b)に示すように1サイクルで並列実行の境界までのユニット(ここでは「実行単位」と呼ぶ)を同時実行する。つまり、各サイクルにおいて並列実行境界情報10が“1”である命令までの命令を並列実行することになる。供給されながら実行されずに残ったユニットは、命令バッファに蓄積され、次のサイクル以降で実行の対象となる。

【0042】つまり、このアーキテクチャでは、固定長のパケット単位で命令を供給しておき、静的に求めた情報を元に、各サイクルにおいて並列度に応じた適切な数のユニットを発行していく、ということになる。この手法をとることにより、通常の固定長命令のVLIW方式で発生していた無動作命令(nop命令)が全く無くなり、コードサイズを削減することができる。

【0043】また、命令内のフォーマット情報11の値によって、2つのユニットを1命令として実行する場合と1つのユニットを1命令として実行する場合がある。この手法をとることにより、命令の指定に多くのビット数を必要とする一部の命令に対してのみ長い命令フォーマットを使用し、他のほとんどの命令については短い命令フォーマットで指定することができるので、さらにコードサイズを削減することができる。具体例については後に述べる。

(プロセッサのハードウェア構成)次に、本プロセッサのハードウェア構成を説明する。

【0044】図3は、本発明に係るプロセッサのハードウェア構成を示すブロック図である。

【0045】本プロセッサは、1サイクルに最大3つの命令を並列実行するプロセッサであり、大きく分けて、命令供給発行部20、解読部30、実行部40から構成される。

【0046】命令供給発行部20は、図示されていない外部メモリから命令群を供給し、解読部30に出力するものであり、命令フェッチ部21、命令バッファ22及び命令レジスタ23からなる。

【0047】命令フェッチ部21は、32ビットのIA(インストラクションアドレス)バス及び64ビットのID(インストラクションデータ)バスを通じて図示されていない外部メモリからユニットのブロックをフェッチし、内部の命令キャッシュに保持すると共に、PC部42から出力されたアドレスに相当するユニット群を命令バッファ22に供給する。

【0048】命令バッファ22は、63ビットのバッファを2個備えており、命令フェッチ部21によって供給されたユニットを蓄積しておくために用いられる。命令バッファ22へは、命令フェッチ部21から64ビット単位でパケットが供給される。ここで、パケットの最上位の1ビットの情報は使用されない。命令バッファ22

に蓄積されたユニットは、命令レジスタ 23 の適切なレジスタに出力される。なお、命令バッファ 22 については、別の図面においてさらに詳細な構成を示している。

【0049】命令レジスタ 23 は、4 個の 21 ビットレジスタからなり、命令バッファ 22 から送られてきたユニットを保持するためのものである。命令レジスタ 23 周辺については、別の図面においてさらに詳細な構成を示している。

【0050】解読部 30 は、命令レジスタ 23 に保持された命令を解読し、その解読結果に応じた制御信号を実行部 40 に出力するものであり、大きく分けて、命令発行制御部 31 と命令デコーダ 32 からなる。

【0051】命令発行制御部 31 は、命令レジスタ 23 の 4 個のレジスタに保持されたユニットに対して、ユニット内の並列実行境界情報 10 とフォーマット情報 11 を参照することによって、2 つのユニットを 1 つの命令として扱うように制御したり、並列実行の境界を越えたユニットについては、そのユニットの発行を無効化したりといった発行に関する制御を行う。なお、命令発行制御部 31 については、別の図面においてさらに詳細な動作説明を行う。

【0052】命令デコーダ 32 は、命令レジスタ 23 に格納された命令群を解読する装置であり、第 1 命令デコーダ 33、第 2 命令デコーダ 34 及び第 3 命令デコーダ 35 からなる。これらのデコーダは、基本的に 1 サイクルに 1 つの命令を解読し、実行部 40 に制御信号を与える。また、命令内に置かれた定数オペランドについては、各命令デコーダから実行部 40 のデータバス 48 に転送される。

【0053】実行部 40 は、解読部 30 での解読結果に基づいて、最大 3 つの命令を並列実行する回路ユニットであり、実行制御部 41、PC 部 42、レジスタファイル 43、第 1 演算部 44、第 2 演算部 45、第 3 演算部 46、オペランドアクセス部 47 及びデータバス 48、49 からなる。

【0054】実行制御部 41 は、解読部 30 での解読結果に基づいて実行部 40 の各構成要素 42～49 を制御する制御回路や配線の総称であり、タイミング制御、動作許可禁止制御、ステータス管理、割り込み制御等の回路を有する。

【0055】PC（プログラムカウンタ）部 42 は、次に解読実行すべき命令が置かれている図示されていない外部メモリ上のアドレスを命令供給発行部 20 の命令フェッチ部 21 に出力する。

【0056】レジスタファイル 43 は、R0～R31 の 32 個の 32 ビットレジスタから構成される。これらのレジスタに格納された値は、第 1 命令デコーダ 33、第 2 命令デコーダ 34 及び第 3 命令デコーダ 35 での解読結果に基づいて、データバス 48 を経由して第 1 演算部 44、第 2 演算部 45 及び第 3 演算部 46 に転送され、

そこで演算が施され、又はそこを単に通過した後に、データバス 49 を経由してレジスタファイル 43 またはオペランドアクセス部 47 に送られる。

【0057】第 1 演算部 44、第 2 演算部 45 及び第 3 演算部 46 は、それぞれ 2 個の 32 ビットデータに対して算術論理演算を行う ALU や乗算器と、シフト演算を行うバレルシフトを内部に有し、実行制御部 41 による制御の下で演算を実行する。

【0058】オペランドアクセス部 47 は、レジスタファイル 43 と図示されていない外部メモリとの間でオペランドの転送を行う回路である。具体的には、例えば、命令内で、オペコードとして“ld”（ロード）が置かれていた場合には、外部メモリに置かれていた 1 ワード（32 ビット）のデータがオペランドアクセス部 47 を経てレジスタファイル 43 の指定されたレジスタにロードされ、また、オペコードとして“st”（ストア）が置かれていた場合には、レジスタファイル 43 の指定されたレジスタの格納値が外部メモリにストアされる。

【0059】上記 PC 部 42、レジスタファイル 43、第 1 演算部 44、第 2 演算部 45、第 3 演算部 46 及びオペランドアクセス部 47 は、図示されるように、データバス 48（L1 バス、R1 バス、L2 バス、R2 バス、L3 バス、R3 バス）及びデータバス 49（D1 バス、D2 バス、D3 バス）で接続されている。なお、L1 バス及び R1 バスはそれぞれ第 1 演算部 44 の 2 つの入力ポートに、L2 バス及び R2 バスはそれぞれ第 2 演算部 45 の 2 つの入力ポートに、L3 バス及び R3 バスはそれぞれ第 3 演算部 46 の 2 つの入力ポートに、D1 バス、D2 バス及び D3 バスはそれぞれ第 1 演算部 44、第 2 演算部 45 及び第 3 演算部 46 の出力ポートに接続されている。

（命令バッファの詳細な構成）次に、命令バッファ 22 の詳細な構成を説明する。

【0060】図 4 は、命令バッファ 22 の詳細な構成を示すブロック図である。命令バッファ 22 は命令バッファ A221 及び命令バッファ B222 の 2 個の 63 ビットのバッファからなり、それぞれ 3 個ずつのユニットを保持することができる。命令バッファ A221 はバッファ A0、A1 及び A2 からなり、それぞれ 1 個ずつのユニットを保持することができる。同様に、命令バッファ B はバッファ B0、B1 及び B2 からなる。

【0061】命令バッファ 22 には、命令フェッチ部 21 から 64 ビット単位でパケットが供給される。ただし、パケットの最上位の 1 ビットの情報は使用されない。この際、命令バッファ A221 と命令バッファ B222 にまたがって供給されることはなく、いずれかのバッファに 63 ビット単位で供給されることになる。命令バッファ 22 に蓄えられたユニットは供給された順序を保っており、その順序やいずれのバッファが有効であるかについては命令バッファ制御部 223 により、状態と

して管理されている。

【0062】命令バッファ制御部223は、毎サイクルバッファ内の有効なユニットを順に命令レジスタ23に転送するため、セクタ224a~224dの制御を行う。この制御により、命令バッファ22内の先頭の4つのユニットが命令レジスタ23に転送される。さらに、命令レジスタ23に転送したユニットの中でどれだけのユニットが発行されずに残ったか、という解読部30の命令発行制御部31からの情報と、命令フェッチ部21から転送されてきたユニットの内いずれのユニットが有効かという情報とを元に、命令バッファ22の状態の更新を行う。

【0063】具体的には、まず命令バッファ22が空の状態、あるパケットの2番目のユニットに分歧した場合には、命令フェッチ部21からそのパケットが供給され、供給されたパケットは命令バッファA221に転送される。そのパケットの先頭のユニットは無効なので、命令バッファ制御部223の制御により、命令バッファ22の状態としてバッファA1及びバッファA2のみが有効な状態となる。

【0064】次のサイクルで命令バッファ22から命令レジスタ23に転送したユニットが全く発行されず、命令フェッチ部21から64ビットの有効なパケットが供給された場合には、そのパケットは命令バッファB222に転送され、命令バッファ22の状態は、バッファA1、A2、B0、B1及びB2が有効な状態となる。

【0065】さらに、次のサイクルでは、命令バッファ22に空きがないので、命令フェッチ部21からの供給は受け付けず、命令レジスタ23へは、順にバッファA1、バッファA2、バッファB0、バッファB1のユニットを転送する。

【0066】このように、命令バッファ22に63ビット単位で空きがある場合にのみ命令フェッチ部21からパケットの供給を行い、供給された順を管理しておき、各サイクルにおいて、供給された順に先頭の4つのユニットを命令レジスタ23に転送していく。

(命令レジスタ23周辺の構成と命令発行制御部31の動作)次に、命令レジスタ23周辺の構成を示し、命令発行制御部31の詳細な動作を説明する。

【0067】図5は、命令レジスタ23周辺の構成を示すブロック図である。図中、破線の矢印は制御信号を表す。

【0068】命令レジスタ23は命令レジスタA231、命令レジスタB232、命令レジスタC233及び命令レジスタD234の4個の21ビットレジスタからなる。命令レジスタ23には、命令バッファ22からユニットが供給されるわけだが、わかりやすくするために命令バッファ22から供給されるユニットの並びであるユニットキュー50という概念を考える。そして、ここでは命令レジスタ23にはユニットキュー50からユニ

ットが供給されると考える。

【0069】図5を見るとわかるように、あるユニットがいずれの命令レジスタ23に転送されるかどうかは、ユニットキュー50での位置(順序)によって一意に決まる。つまり、ユニット1は命令レジスタA231へ、ユニット2は命令レジスタB232へ転送されることになる。これにより、ユニットキュー50から命令レジスタ23への転送を行う際に、図13の従来例において存在したようなユニットの選択を行うセクタが不要となり、ハードウェアが単純化されており、遅延も最小限に抑えられている。

【0070】図中33~35の各命令デコーダは、21ビットのユニットを入力とし、それを解読して、そのユニットが構成する命令の動作に関する制御信号を実行制御部41に出力するとともに、ユニット内に配置された定数オペランドを出力する。図1の命令フォーマットからわかるように、42ビット命令を構成する2つのユニットのうち、2番目のユニットには定数オペランドの一部しか配置されない。つまり、このユニットにはオペコードが存在しないため、命令デコーダに入力する必要がない。そこで、各命令の定数オペランドは、図5に示されるように、命令デコーダが出力したユニット内の定数と、命令レジスタから無条件に直接転送された定数とを連結したものということになる。図5の60~62が各命令の定数オペランドである。

【0071】また、各命令デコーダには、制御信号として1ビットの無動作命令フラグが入力される。このフラグを“1”にセットすると、そのデコーダは出力として無動作命令を出力する。つまり、無動作命令フラグをセットすることにより、その命令デコーダの命令としてのデコードを無効化することができる。

【0072】ここで、命令レジスタ23に格納されたユニットを組み合わせる命令として発行する制御を行う命令発行制御部31の動作について説明する。

【0073】命令発行制御部31は、命令レジスタA231及び命令レジスタB232に格納された各ユニットの並列実行境界情報10とフォーマット情報11を参照することにより命令デコーダの制御を行う。

【0074】まず、これらの情報から、命令レジスタ23に格納されたユニットの内どこまでをこのサイクルで発行するのかを求める。そして、どれだけのユニットが発行されずに残ったのかの情報を命令バッファ22内の命令バッファ制御部223に伝達する。

【0075】次に命令デコーダ32を制御し、このサイクルで発行される命令についてのみ解読を行うように制御する。図5からわかるように、命令としてデコードされる可能性のあるユニットは、命令レジスタA231、命令レジスタB232及び命令レジスタC233に格納されたユニットのみである。そこで、ユニット内の情報を参照して、これらのユニットの中で、42ビット命令

の2ユニット目にあたるものや発行されずに残るものに関しては、そのユニットの命令としてのデコードを無効化する。42ビット命令の2ユニット目にあたるユニットは、直前のユニットが構成する命令の定数オペランドの一部として直接出力される。

【0076】具体的には、命令レジスタA231のユニット（ユニット1）のフォーマット情報11が“1”のときには、ユニット1と命令レジスタB232のユニット（ユニット2）とを連結して42ビット命令となるので、ユニット2の命令としてのデコードを無効化する、すなわち第2命令デコーダ34の無動作命令フラグを“1”にセットする。図5において、命令発行制御部31から第2命令デコーダ34への破線がこの動作に相当する。ユニット2は、ユニット1が構成する命令の定数オペランド60の一部として直接出力される。

【0077】また、ユニット1のフォーマット情報11が“0”、ユニット2のフォーマット情報が“1”の時は、ユニット2と命令レジスタC233のユニット（ユニット3）とを連結して42ビット命令となるので、ユニット3の命令としてのデコードをキャンセルする、すなわち第3命令デコーダ35の無動作命令フラグを“1”にセットする。図5において、命令発行制御部31から第3命令デコーダ35への破線がこの動作に相当する。ユニット3は、ユニット2が構成する命令の定数オペランド61の一部として直接出力される。

【0078】このように、フォーマット情報11を参照することにより、必要に応じて命令デコーダの無動作フラグを設定し、命令としてのデコードを無効化する。

【0079】それから、ユニット1の並列実行境界情報10が“1”、フォーマット情報11が“0”のときは、このサイクルではユニット1までしか発行されないもので、ユニット2とユニット3の命令としてのデコードを無効化する、すなわち第2命令デコーダ34と第3命令デコーダ35の無動作命令フラグを共に“1”にセットする。図5において、命令発行制御部31から第2命令デコーダ34と第3命令デコーダ35への破線がこの動作に相当する。

【0080】また、ユニット1の並列実行境界情報10が“0”、ユニット2の並列実行境界情報10が“1”、フォーマット情報11が共に“0”のときは、このサイクルではユニット2までしか発行されないもので、ユニット3の命令としてのデコードを無効化する、すなわち第3命令デコーダ35の無動作命令フラグを共に“1”にセットする。図5において、命令発行制御部31から第3命令デコーダ35への破線がこの動作に相当する。

【0081】このように、並列実行境界情報10を参照することにより、必要に応じて命令デコーダの無動作フラグを設定し、命令としてのデコードを無効化する。

【0082】以上のような命令発行制御を実現する命令

発行制御部31とその周辺回路の構成を図6に示す。

【0083】前述のように命令発行制御部31は命令レジスタA231及び命令レジスタB232に格納されたユニットの並列実行境界情報10とフォーマット情報11を参照し、第2命令デコーダ34及び第3命令デコーダ35の命令としてのデコードを無効化するかどうかを決定する無動作命令フラグとなる制御信号を出力する。

【0084】図6のような回路構成をとることにより、第2命令デコーダ34は、命令レジスタA231に格納されたユニットの並列実行境界情報10が“1”であるか、またはそのユニットのフォーマット情報11が“1”であるときに無効化される。また、第3命令デコーダ35は、命令レジスタA231に格納されたユニットもしくは命令レジスタB232に格納されたユニットの並列実行境界情報10が“1”であるか、または命令レジスタB232に格納されたユニットのフォーマット情報11が“1”であるときに無効化される。

【0085】このように、図1に示したような命令フォーマットをとり、図6に示したような単純な回路を用意するだけで、必要最低限の情報を参照して高速な命令発行制御を行うことができる。

【0086】以上で述べたような命令発行制御の方法をとることにより、1サイクルで同時発行可能な命令の命令長の組み合わせに多少の制限が生じる。本プロセッサで同時発行可能な命令の命令長の組み合わせを図7に示す。

【0087】図7を見るとわかるように、本プロセッサでは、ユニットの並びの先頭から3つ目までのユニットについてのみ命令としてデコードすることができる。つまり、図中（a）～（h）のパターンについて発行することができる。最大で4つのユニットを同時に発行することになる。ただし、4つのユニットを発行するパターンの内、図中（i）、（j）のパターンについては同時発行することができない。

（従来の命令発行制御方法との比較）ここで、本実施形態のプロセッサと本発明によらない従来のプロセッサとの比較を行う。

【0088】まず、図13に示した従来例において、ユニットキュー50から命令レジスタへの転送において、セレクトによる遅延が問題となっていたが、本発明のプロセッサでは、ユニットキュー50内のユニットと各命令レジスタが一对一に対応しているため、図13において存在していたセレクト51a～51dが不要となり、上記遅延の問題が解決されている。

【0089】また、図13の構成では、並列度が増して転送すべき命令レジスタが増加していくに従って、セレクトが増加し遅延がさらに大きくなっていくのに対して、本発明のプロセッサでは、ユニットキュー50と命令レジスタの対応は一对一なので、遅延が大きくなることはない。

【0090】一方、この可変長命令方式をスーパースカラ方式にて並列実行を行うプロセッサに適用したものが提案されている。例えば、論文 The Approach to Multiple Instruction Execution in the GMICRO/400 Processor (PROCEEDINGS, The Eighth TRON Project Symposium (International), 1991参照) にて開示されている GMICRO/400 がある。この技術は、図 14 の概念をとりながらもハードウェアを削減するために制限を設けている。

【0091】図 15 は、GMICRO/400 で採用されている命令発行制御方法をとった場合の命令レジスタ周辺の構成を示すブロック図である。図 15 において、破線は制御信号を表し、54a 及び 54b は命令内に指定された定数オペランドを表す。命令デコーダは、入力された命令を解釈し、その結果その命令の実行を制御する信号を実行制御部に出力すると共に、命令内に指定された定数オペランドを出力する。

【0092】GMICRO/400 の命令発行制御方法では、ユニット 1 とユニット 2 を連結したものの、ユニット 2 及びユニット 3 をそれぞれ一旦デコードしておき、第 1 命令デコーダ 53i の解釈によって 1 番目の命令が 1 ユニット長の命令なのか 2 ユニット長の命令なのかが判明した時点で、セクタ 51g およびセクタ 51h を制御することにより、第 2 命令デコーダ 53j もしくは第 3 命令デコーダ 53k の解釈結果を選択して使用する。

【0093】図 15 を見るとわかるように、GMICRO/400 では、図 14 の構成に対して、同時実行可能な命令数を 3 から 2 に減らすことにより、第 4 命令デコーダ 53g と第 5 命令デコーダ 53h を削除している。また、第 2 命令デコーダ 53j と第 3 命令デコーダ 53k は、入力ビット幅を 1 ユニット長とし、ハードウェア削減を図っている。しかし、これによって、同時実行される 2 番目の命令は 1 ユニット長の命令のみという制限が発生する。

【0094】以上のようなハードウェア削減を図っても、2 命令同時発行を可能にするために 3 つの命令デコーダを必要としており、依然としてハードウェア量が多いという問題点がある。

【0095】また、図 15 の構成では、第 1 命令デコーダ 53i の解釈が完了するまでセクタ 51g、51h の制御を決定することができない。このセクタの制御が決定するまで、2 番目の命令として第 2 命令デコーダ 53j と第 3 命令デコーダ 53k のいずれの解釈結果を用いるかを決定できず、オペランドとなるレジスタの格納値の読み出しを開始できない。オペランドとなる可能性のある全てのレジスタの格納値を先行的に読み出しおき、それを選択して使用方法も考えられるが、レジスタファイルの読み出しポート数が増加するため実用的ではない。このように、図 15 の構成では読み出すレ

ジスタを確定するまでの遅延が大きくなる。実際、GMICRO/400 では命令解釈ステージの直後のステージでは演算の実行は行わず、オペランドを読み出すステージとしている。

【0096】さらに、図 15 の構成にて並列度が増して同時発行可能なユニット数が増加していくと、セクタの数が増し、制御が複雑化するという問題点がある。

【0097】以上に述べたように、スタティックスケジューリングによってさらなる並列化を実現し、性能向上を図ることができるが、コードサイズが大きくなるという問題点がある。また、コードサイズを削減する手段として可変長命令方式があるが、ハードウェアが複雑になるという問題点がある。

【0098】そして、図 15 に示した GMICRO/400 の例においては、命令デコーダを 3 つ用意しても最大 2 命令しが同時実行できないのに対して、本発明のプロセッサの命令発行制御方法を用いると、3 つの命令デコーダにて最大 3 命令を同時実行することができる。逆に、本発明において、最大 2 命令を同時発行する構成を想定した場合、2 個の命令デコーダにて構成することができる。具体的な命令レジスタ周辺の構成は図 16 のようになる。これにより、ハードウェアを削減することができる。

【0099】また、図 15 の構成では、第 1 命令デコーダ 53i の解釈が完了するまでセクタ 51g、51h の制御を決定することができず、2 番目の命令として第 2 命令デコーダ 53j と第 3 命令デコーダ 53k のいずれの解釈結果を用いるかを決定できない。そのため、オペランドとなるレジスタを確定するまでの遅延が大きくなる。これに対して、本発明のプロセッサの構成では、他のデコーダの解釈結果を待たずに、オペランドとなるレジスタを確定することができるため、解釈ステージの前半にオペランドとなるレジスタの読み出しを開始することができる。その結果、解釈ステージの完了時点で、オペランドとなるレジスタの読み出しも完了させておくことができる。これによって、解釈ステージの直後のステージで演算を実行することができ、実行効率を高めることができる。

【0100】さらに、本発明の構成では、同時発行可能なユニット数が増加しても単純にデコーダの数を増していけばよいのに対して、図 15 の構成では、同時発行可能なユニット数が増加していくと、セクタの数が増して制御が複雑化するという問題点がある。

【0101】それから、命令フォーマットの違いによる差異として次のものがある。本発明では図 1 のように、2 ユニット長の命令の 2 番目のユニットには定数オペランドの一部のみが配置されるため、図 5 のように 2 番目のユニットは命令デコーダには入力されずに直接オペランドとして出力される。このため、すべての命令デコーダは 1 ユニット長の命令を解釈するだけでよい。これに

対して、GMICRO/400では、2ユニット長の命令の2番目のユニットにもオペコードが配置されるため、図15の構成で第1命令デコーダ53iは2ユニット長の命令を解読する必要があり、本発明の構成に比べてハードウェアが増加している。

(プロセッサの動作) 次に、具体的な命令を解読実行した場合の本実施形態のプロセッサの動作について説明する。

【0102】図8は、32ビットの定数を扱う処理の一例を示すフローチャートである。本図に示されている処理は、32ビットの定数“0x87654321”をレジスタR1に転送し(ステップS100)、レジスタR5の格納値をレジスタR0に転送し(ステップS101)、レジスタR0の格納値にレジスタR1の格納値を加え(ステップS102)、レジスタR3の格納値にレジスタR2の格納値を加え(ステップS103)、レジスタR0の格納値をメモリ内のレジスタR4の格納値で示されるアドレスに格納し(ステップS104)、レジスタR0の格納値をレジスタR6に転送し(ステップS105)、最後にレジスタR3の格納値をレジスタR7に転送する(ステップS106)というものである。

【0103】図9は、図8に示された処理を本プロセッサに行わせるプログラムの実行コードの例と実行イメージを示す図である。

【0104】このプログラムは、7個の命令で構成されており、命令供給単位としては3個のパケット70~72から構成されている。各命令の処理内容は、実行コードの各フィールドに置かれたニーモニックで表現されている。具体的には、ニーモニック“mov”は、定数及びレジスタの格納値のレジスタへの転送を表し、ニーモニック“add”は、定数及びレジスタの格納値とレジスタの格納値との加算を表し、ニーモニック“st”は、レジスタの格納値のメモリへの転送を表している。

【0105】なお、定数は16進数で表現されている。また、“Rn(n=0~31)”はレジスタファイル43の中の一つのレジスタを示す。そして、各命令の並列実行境界情報10とフォーマット情報11についても“0”又は“1”で示してある。

【0106】図9(b)を用いて、図8に示された処理における各実行単位ごとの本プロセッサの動作を説明する。

(実行単位1) パケット70がメモリから供給され、パケット70内のユニットが順に命令レジスタ23に転送される。次に、命令発行制御部31が各ユニットの並列実行境界情報10とフォーマット情報11を参照して発行を制御する。具体的には、1番目のユニットのフォーマット情報11が“1”であるので、1番目のユニットと2番目のユニットを連結して1つの命令として扱う。つまり、第2命令デコーダ34の無動作命令フラグを“1”にセットして、命令としてのデコードを無効化する。

また、1番目のユニットの並列実行境界情報10が“0”であり、3番目のユニットの並列実行境界情報10が“1”であるので、3番目のユニットまでの2個の命令を発行する。供給されたすべてのユニットを発行するため、命令バッファ22にはユニットは蓄積されない。

【0107】実行部40では、レジスタR1に定数“0x87654321”が転送され、レジスタR5の格納値がレジスタR0に転送される。

(実行単位2) パケット71がメモリから供給され、パケット71内のユニットが順に命令レジスタ23に転送される。3個のユニット共フォーマット情報11が“0”であるので、いずれのユニットも21ビット命令となる。また、1番目のユニットの並列実行境界情報10が“0”であり、2番目のユニットの並列実行境界情報10が“1”であるので、2番目のユニットまでの2個の命令を発行する。3番目のユニットは、発行されずに残ったので命令バッファ22に蓄積される。

【0108】実行部40では、レジスタR0の格納値にレジスタR1の格納値が加えられてレジスタR0に格納され、レジスタR3の格納値にレジスタR2の格納値が加えられてレジスタR3に格納される。

(実行単位3) パケット72がメモリから供給され、命令バッファ22に蓄積されていた1個のユニットとパケット72内の2個のユニットとが順に命令レジスタ23に転送される。3個のユニット共フォーマット情報11が“0”であるので、いずれのユニットも21ビット命令となる。また、1番目のユニットの並列実行境界情報10と2番目のユニットの並列実行境界情報が“0”であり、3番目のユニットの並列実行境界情報10が“1”であるので、3番目のユニットまでの3個の命令を発行する。これで、供給されたユニットはすべて発行されたことになる。

【0109】実行部40では、レジスタR0の格納値がメモリ内のレジスタR4の格納値で示されるアドレスに転送され、レジスタR0の格納値がレジスタR6に転送され、レジスタR3の格納値がレジスタR7に転送される。

【0110】以上のようにして、本プロセッサにおいて図8に示した処理を行うプログラムは3つの実行単位で実行される。実行コードは、42ビット命令が1個と21ビット命令が6個で構成されていたので、コードサイズは168ビットである。

(従来の固定長VLIW方式のプロセッサとの比較) 次に、図8に示した処理を、従来技術の1つとして挙げた命令長が固定のVLIW方式のプロセッサに行わせた場合を仮定して、本発明に係るプロセッサの場合と比較する。

【0111】毎サイクル固定長の命令を固定個数発行する単純なVLIW方式では、32ビットの定数を転送す

る命令を1命令で指定できるような命令長にすると、非常にコードサイズが大きくなってしまいうため、命令長は32ビットとし、32ビットの定数の転送は16ビットずつ2命令に分けて行うことにする。

【0112】図10は、図8に示された処理を、命令長が32ビット固定のVLIW方式のプロセッサに行わせるプログラムの実行コードの例と実行イメージを示す図である。

【0113】このプログラムは、4個のパケット73～76から構成されている。各命令の処理内容は、図9に示したコードと同様に、各フィールドに置かれたニーモニックで表現されている。ただし、ニーモニック“sethi”は、16ビットの定数をレジスタの上位16ビットに格納することを表し、ニーモニック“setlo”は、16ビットの定数をレジスタの下位16ビットに格納することを表し、ニーモニック“nop”は、何もしない命令であることを表している。

【0114】図10(a)の実行コードと同図(b)の実行イメージとを比較するとわかるように、VLIW方式では、各サイクル供給された命令がそのまま発行される。つまり、毎サイクル32ビット命令が3個発行されることになる。並列実行可能な命令が存在しない場合は、あらかじめソフトウェアで“nop”命令を挿入しておく必要がある。そのため、この例でも4個の“nop”命令が挿入されて、コードサイズは32ビット命令が12個なので384ビットとなっており、本発明に係るプロセッサの場合のコードサイズよりも大幅に大きいものになっている。

【0115】また、32ビットの定数のレジスタへの転送を2命令に分けて行っているために新たな依存関係が生じ、実行単位の数に4つとなっている。どのような命令並べ替えを行っても実行単位の数減らすことはできない。これによって、本発明に係るプロセッサの場合に比べて実行サイクル数が1サイクル増加する。

(従来の並列実行境界の情報を固定長命令内に持つプロセッサとの比較) 次に、図8に示した処理を、従来技術の1つとして挙げた命令長が固定であり並列実行の境界であるか否かの情報を命令内に持つ方式のプロセッサに行わせた場合を仮定して、本発明に係るプロセッサの場合と比較する。

【0116】この方式では、命令長が32ビットのモデルと40ビットのモデルを考える。命令長が32ビットのモデルでは、図10のVLIW方式の場合と同様に、32ビットの定数のレジスタへの転送は2命令に分けて行う。それに対して命令長が40ビットのモデルでは、32ビットの定数のレジスタへの転送を含むすべての種類の演算を1命令で指定することができる。

【0117】図11は、図8に示された処理を、命令長が32ビット固定であり並列実行の境界の情報を命令内に持つ方式のプロセッサに行わせるプログラムの実行コ

ードの例と実行イメージを示す図である。

【0118】このプログラムは、8個の命令で構成されており、命令供給単位としては3個のパケット77～79から構成されている。各命令の処理内容は、実行コードの各フィールドに置かれたニーモニックで表現されている。32ビットの定数のレジスタへの転送は、図10の命令長32ビット固定のVLIW方式の場合と同様に16ビットずつ2個の命令に分けて行う。

【0119】図11を見ると分かるように、このモデルでも図10のVLIW方式の場合と同様に32ビットの定数のレジスタへの転送を2命令に分けて実行しているため、新たな依存関係が生じ、実行サイクル数が本発明に係るプロセッサの場合に比べて1サイクル増加している。

【0120】コードサイズに関しては、“nop”命令の挿入が発生しないため、図10のVLIW方式の場合のコードサイズから丁度“nop”命令の分だけ削減されており、32ビット命令が8個で256ビットとなっている。しかし、依然として本発明に係るプロセッサの場合のコードサイズに比べると大きい。

【0121】次に、命令長を40ビット固定としたモデルとの比較を行う。図12は、図8に示された処理を、命令長が40ビット固定であり並列実行の境界の情報を命令内に持つ方式のプロセッサに行わせるプログラムの実行コードの例と実行イメージを示す図である。

【0122】このプログラムは、7個の命令で構成されており、命令供給単位としては3個のパケット80～82から構成されている。各命令の処理内容は、実行コードの各フィールドに置かれたニーモニックで表現されている。32ビットの定数のレジスタへの転送についても、1命令で指定することが可能である。

【0123】図12を見ると分かるように、このモデルでは32ビットの定数のレジスタへの転送を1命令で指定することができるため、実行単位数は3つであり、実行サイクル数は本発明に係るプロセッサの場合と同じである。

【0124】命令数は本発明に係るプロセッサの場合と同じだが、本発明に係るプロセッサの場合は長いビット数を必要としない命令については21ビット命令で指定できるのに対し、このモデルではすべての命令を40ビット命令で指定する必要があるため、コードサイズは40ビット命令が7個で280ビットとなっており、本発明に係るプロセッサの場合に比べて大きくなっている。

【0125】以上、本発明に係るプロセッサについて、実施形態に基づいて説明したが、本発明はこれらの実施形態に限られないことは勿論である。即ち、

(1) 上記実施の形態では、静的なスケジューリングを前提としていたが、本発明はこれに限定されるものではない。つまり、スーパースカラ方式のように動的なスケジューリングを行うプロセッサにも適用することができ

る。この場合は、命令フォーマット内の並列実行境界情報を無くし、解読部の中に動的に並列実行可能か否かを検出する並列実行可否検出装置を持たせ、本実施形態において命令発行制御部にて並列実行境界情報を参照して行っていた制御を、並列実行可否検出装置の出力を参照して行えばよい。このような構成にしても、可変長命令方式においてハードウェアを簡単化できるという本発明の有意性は保たれる。

(2) 上記実施の形態では、3 個の命令を同時実行するように構成していたが、本発明はこの同時実行命令数に限定されるものではない。例えば、2 個の命令を同時実行する構成にしてもよい。この場合は、解読部と命令レジスタ周辺の構成を図 16 のブロック図に示すように変更し、実行部の演算器の構成を適宜変更すればよい。

(3) 上記実施の形態では、図 1 の命令フォーマットからわかるように、ユニット 1 個または 2 個にて 1 個の命令を構成していたが、本発明はこのユニット数に限定されるものではない。つまり、3 個以上のユニットを連結して 1 個の命令を構成するような命令フォーマットを定義してもよい。例えば、1 ~ 4 個の単位命令にて命令を構成する場合には、命令内のフォーマット情報を 2 ビットにすればよい。

(4) 上記実施の形態では、図 1 の命令フォーマットからわかるように、ユニット 1 個または 2 個にて 1 個の命令を構成していたが、必ずしもユニット単体で構成される命令が存在する必要はない。例えば、1 命令が 2 個または 3 個のユニットから構成されるときもよい。この場合には、命令レジスタと命令デコーダおよび定数オペランドを結ぶ配線を変更すればよい。

(5) 上記実施の形態では、図 1 の命令フォーマットからわかるように、命令内に並列実行の境界であるか否かの情報を持たせていたが、この情報は必ずしも必要ではない。つまり、命令内にはフォーマットに関する情報のみをもち、並列実行可能な命令が存在しない場合には“nop”命令を配置するという方法をとってもよい。この場合においても、各命令の指定に必要な長さの命令フォーマットにて命令を指定することができるという本発明の有意性が保たれる。

(6) 上記実施の形態では、図 1 の命令フォーマットからわかるように、42 ビット命令を構成する 2 つのユニットのうち 2 番目のユニットには定数オペランドの一部のみを配置するようになっていたが、このユニットにオペコードを配置しても構わない。そのためには、図 5 において直接定数オペランドの一部として出力していたユニットを命令デコーダへ入力するように変更し、命令デコーダの入力ビット幅を増加させればよい。

(7) 上記実施の形態では、命令バッファの構成として図 4 に示すものとしたが、本発明はこの構成およびバッファのサイズに限定されるものではない。例えば、一本の単純なキュー構造の命令バッファを用いてもよい。

【0126】

【発明の効果】以上の説明から明らかなように、本発明のプロセッサは、一語が複数でかつ所定数の語素からなる複合命令を実行する VLIW 方式のプロセッサであって、実行の単位となる単位命令が一または複数の前記語素から構成され、かつ前記単位命令を構成する前記語素の数が少なくとも二通りあることを特徴とする。

【0127】これによって、各命令の指定に必要なビット数に応じた命令長の命令フォーマットにて命令を指定することができ、コードサイズを削減することができる。

【0128】また本発明のプロセッサは、一語が複数でかつ所定数の語素からなる複合命令を実行する VLIW 方式のプロセッサであって、実行の単位となる単位命令が一または複数の前記語素から構成され、かつ前記単位命令を構成する前記語素の数が少なくとも二通りあり、前記複合命令中の前記単位命令と該単位命令に後続する単位命令とが並列に実行できるか否かを示す並列実行情報が、前記複合命令内に明示的に含まれることを特徴とする。

【0129】これによって、各命令の指定に必要なビット数に応じた命令長の命令フォーマットにて命令を指定できると共に、一般の VLIW などと並列実行可能な命令が存在しない場合に挿入される無動作命令の挿入が不要となり、コードサイズを削減することができる。

【0130】また本発明のプロセッサは、一語が複数の語素からなる複合命令を実行し、実行の単位となる単位命令が一または複数の前記語素から構成され、かつ前記単位命令を構成する前記語素の数が少なくとも二通りあるプロセッサであって、記憶装置から前記複合命令を読み出す複合命令読み出し手段と、前記複合命令読み出し手段により読出された前記複合命令中の前記語素から、いずれかの前記語素内に含まれ前記単位命令を構成する前記語素の数に関する語数情報に基づいて、前記命令単位を組み立てて解読する解読手段とを備え、前記単位命令は、第 1 の数の前記語素から構成される単位命令と、前記第 1 の数より大きい第 2 の数の前記語素から構成される単位命令とがあり、前記第 2 の数の前記語素から構成される単位命令の内にあって、前記第 1 の数の前記語素から構成される単位命令に相当する部分からはみ出す部分の語素には、命令中の数値に関する記述のみが配置されることを特徴とする。

【0131】これによって、数値に関する記述に関しては命令デコーダに入力せず、直接定数オペランドとして出力することができ、命令デコーダを簡単化することができる。これにより、ハードウェアを単純化することができる。

【0132】また本発明のプロセッサは、一語が複数の語素からなる複合命令を実行し、実行の単位となる単位

命令が一または複数の前記語素から構成され、かつ前記単位命令を構成する前記語素の数が少なくとも二通りあるプロセッサであって、記憶装置から前記複合命令を読み出す複合命令読み出し手段と、前記複合命令読み出し手段により読出された前記複合命令中の前記語素から、いずれかの前記語素内に含まれ前記単位命令を構成する前記語素の数に関する語数情報に基づいて、前記命令単位を組み立てて解読する解読手段とを備え、前記解読手段は、前記語素の一定数ごとに区切られた前記複合命令をそれぞれ解読する複数の単位解読手段と、前記語数情報に基づいて前記単位解読手段のいくつかの解読を無効化する解読制御手段とから構成されることを特徴とする。

【0133】これによって、命令の供給部から命令の発行部への単位命令の転送の際のセクタによる遅延を解消し、またハードウェアを単純化することができる。よって、命令長を可変とすることによりハードウェアが複雑になり、遅延が増大して性能が低下する、という問題を克服している。さらに、この命令発行制御方法では、並列度が増して同時発行可能な命令数が増えても、単位命令の命令レジスタへの転送の遅延が増加しないというメリットがある。

【0134】さらに、GMICRO/400で採用されている構成に対してデコーダの数を減らすことができる。また、解読ステージの前半にオペランドの読み出しを開始することができるため、解読ステージが完了する時点でオペランドの読み出しを完了させておくことができ、実行効率を高めることができる。

【図面の簡単な説明】

【図1】本発明の実施形態に係るプロセッサが実行する命令の構成を示す図

【図2】同プロセッサにおける命令の供給と発行の概念を示す図

【図3】同プロセッサのハードウェア構成を示すブロック図

【図4】同プロセッサの命令バッファ22の詳細な構成を示すブロック図

【図5】同プロセッサの命令レジスタ23周辺の構成を示すブロック図

【図6】同プロセッサの命令発行制御部31とその周辺の回路構成を示す図

【図7】同プロセッサが同時発行可能な命令群の命令長の組み合わせを示す図

【図8】32ビットの定数を扱う処理の一例を示すフローチャート

【図9】図8に示された処理を図3のプロセッサに行わせるプログラムの実行コードの例と実行イメージを示す図

【図10】図8に示された処理を命令長が32ビット固定のVLIW方式の従来のプロセッサに行わせるプログ

ラムの実行コードの例と実行イメージを示す図

【図11】図8に示された処理を、命令長32ビット固定で命令内に並列実行境界の情報を持たせる方式の従来のプロセッサに行わせるプログラムの実行コードの例と実行イメージを示す図

【図12】図8に示された処理を、命令長40ビット固定で命令内に並列実行境界の情報を持たせる方式の従来のプロセッサに行わせるプログラムの実行コードの例と実行イメージを示す図

【図13】従来のプロセッサにおける命令レジスタ周辺の構成を示すブロック図

【図14】従来のプロセッサにおける命令レジスタ周辺の構成を示すブロック図

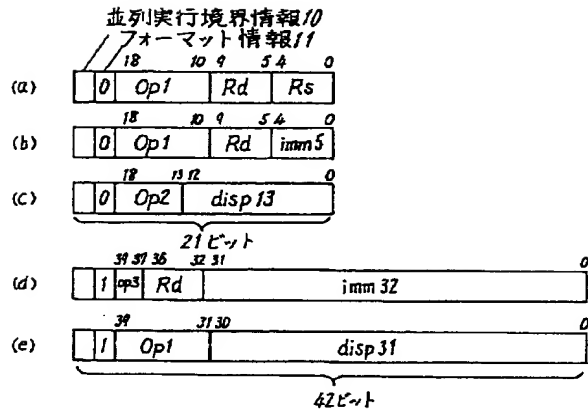
【図15】従来のプロセッサの一例であるGMICRO/400における命令レジスタ周辺の構成を示すブロック図

【図16】本発明の別の実施形態のプロセッサにおける命令レジスタ23周辺の構成を示すブロック図

【符号の説明】

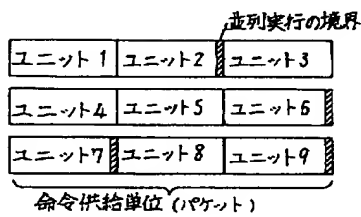
- 10 並列実行境界情報
- 11 フォーマット情報
- 20 命令供給発行部
- 21 命令フェッチ部
- 22 命令バッファ
- 23 命令レジスタ
- 30 解読部
- 31 命令発行制御部
- 32 命令デコーダ
- 33 第1命令デコーダ
- 34 第2命令デコーダ
- 35 第3命令デコーダ
- 40 実行部
- 41 実行制御部
- 42 PC部
- 43 レジスタファイル
- 44 第1演算部
- 45 第2演算部
- 46 第3演算部
- 47 オペランドアクセス部
- 48、49 データバス
- 50 ユニットキュー
- 221 命令バッファA
- 222 命令バッファB
- 223 命令バッファ制御部
- 224 a~224 d セレクタ
- 231 命令レジスタA
- 232 命令レジスタB
- 233 命令レジスタC
- 234 命令レジスタD

【図 1】

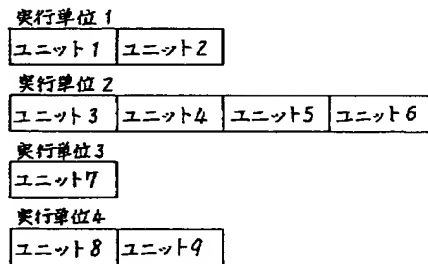


【図 2】

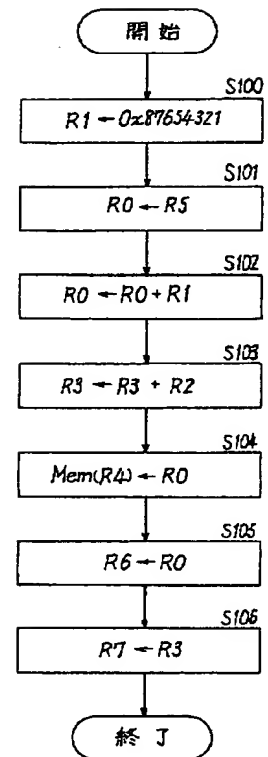
(a) 実行コード



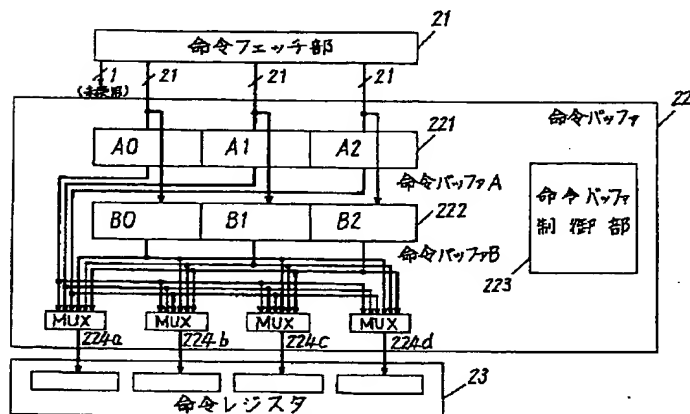
(b) 実行イメージ



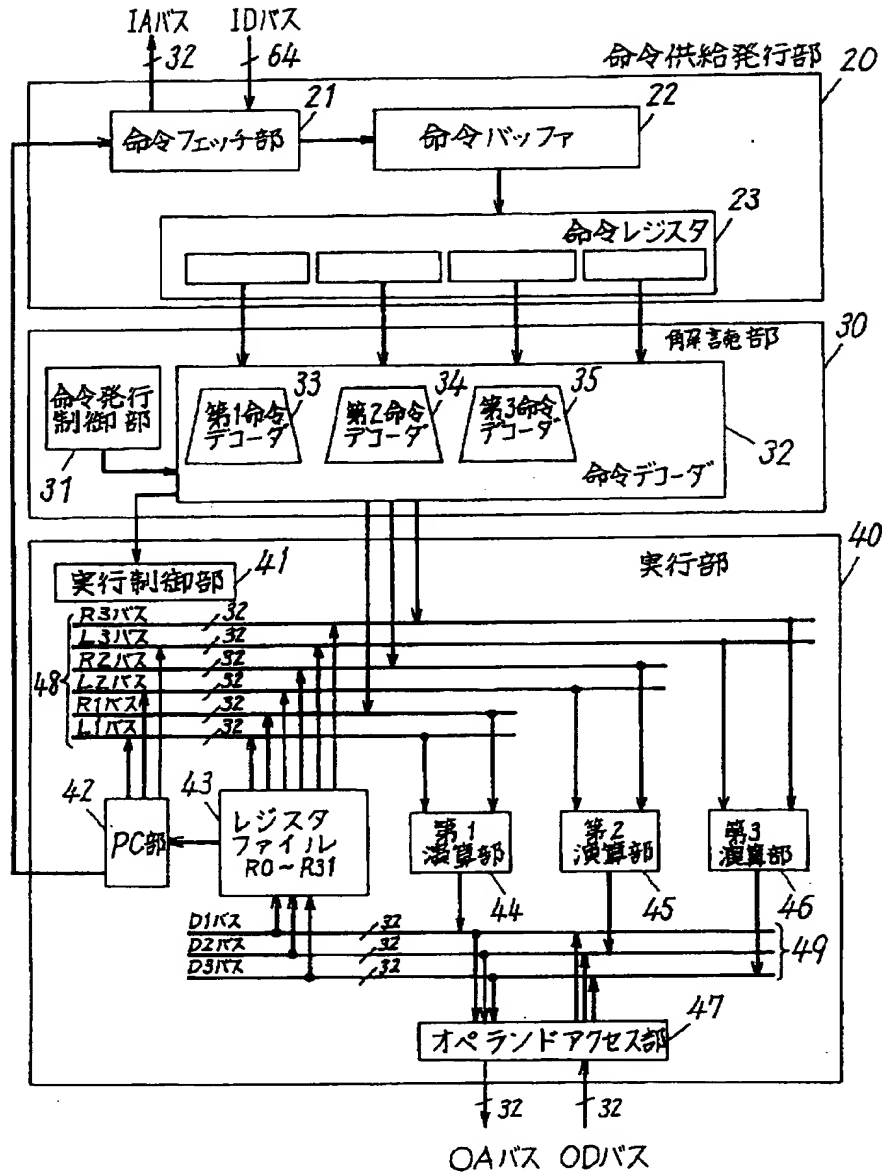
【図 8】



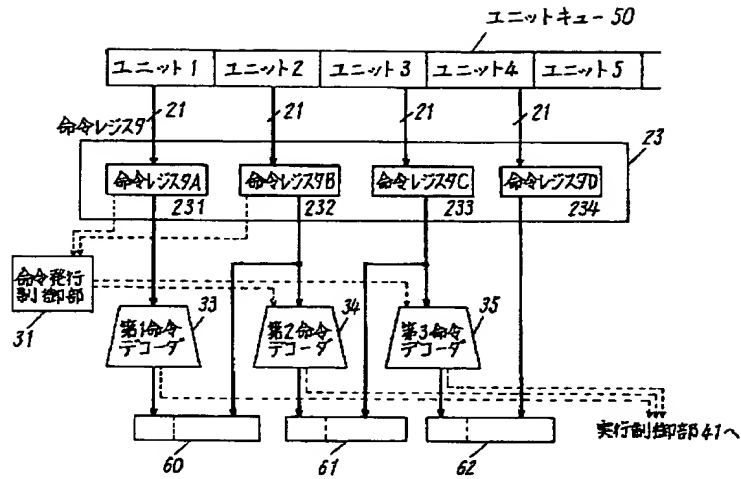
【図 4】



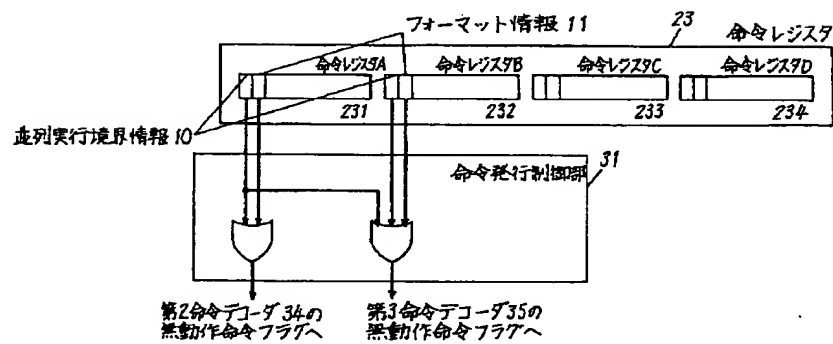
【図3】



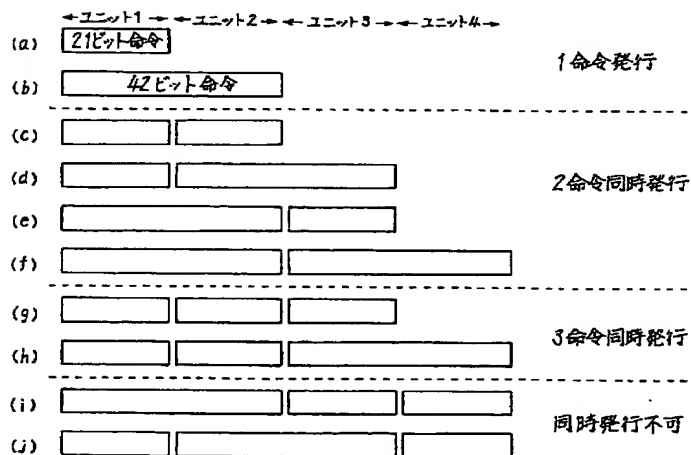
【図5】



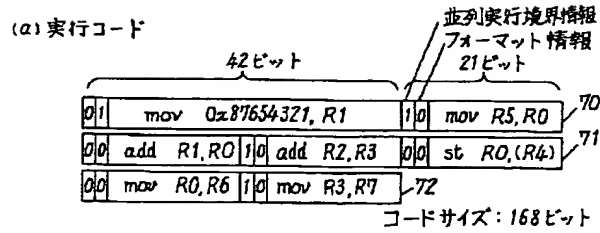
【図6】



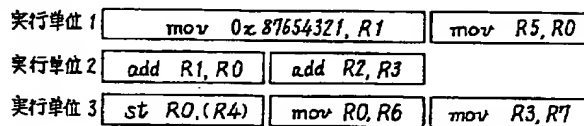
【図7】



【図 9】

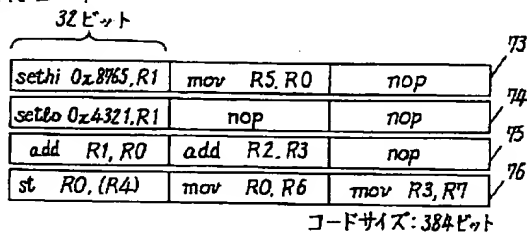


(b) 実行イメージ

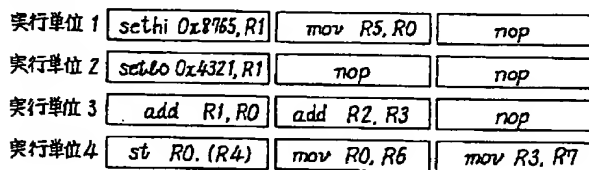


【図 10】

(a) 実行コード

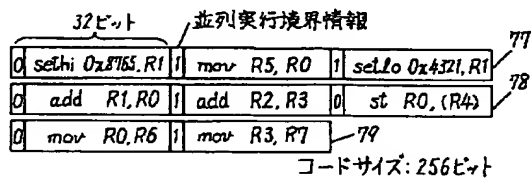


(b) 実行イメージ

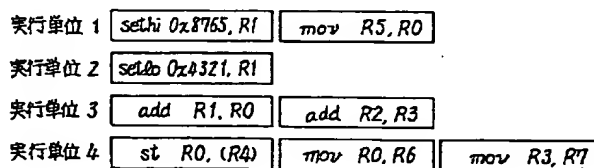


【図 11】

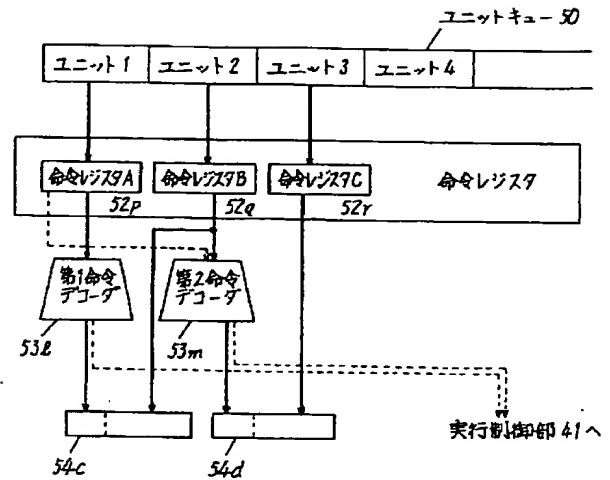
(a) 実行コード



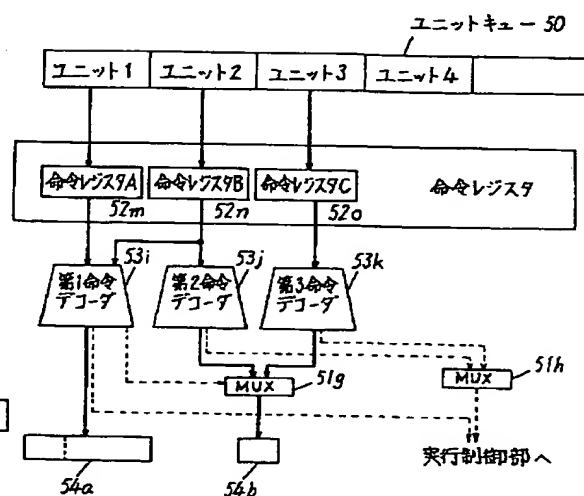
(b) 実行イメージ



【図 16】

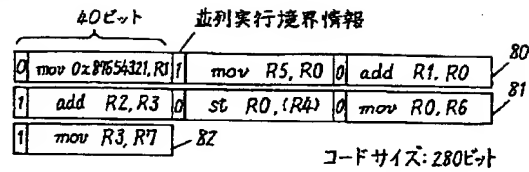


【図 15】

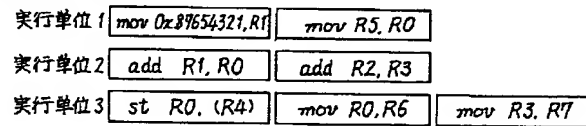


【図 12】

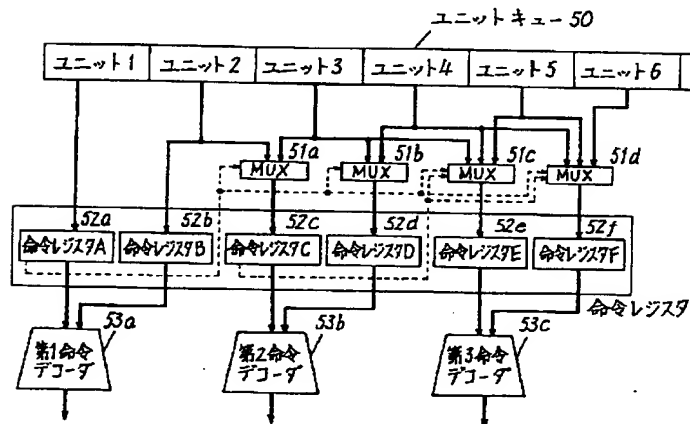
(a) 実行コード



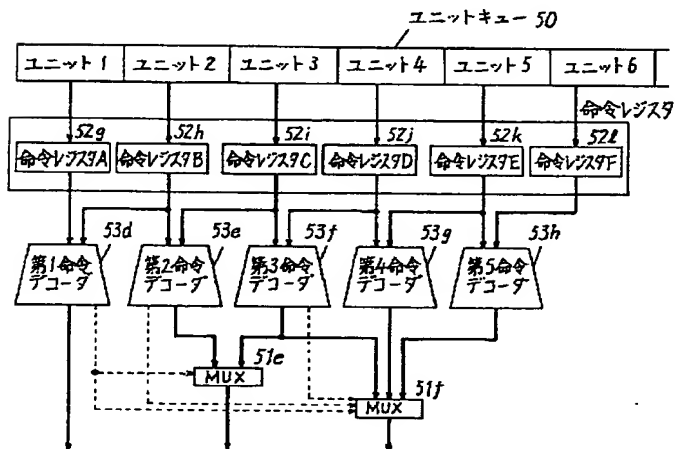
(b) 実行イメージ



【図 13】



【図 14】



フロントページの続き

(72) 発明者 高山 秀一

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(72) 発明者 小谷 謙介

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

THIS PAGE BLANK (USPTO)